Automated validation of variable flow fuel pump diagnosis in virtual environment

Andrea Domingues
Claudio Labate
João Pedro Ceccatto
Vilmara Silva
Ford Motor Company Brasil Ltda.

ABSTRACT

Emission control systems have specific legislation called On Board Diagnostics (OBD). To comply with emission requirements, a significant part of the engine control module (ECU) software is related to OBD, and various validation tests are required for all its functions. These tests are mostly conducted on prototype vehicles or complex test benches such as Hardware-in-the-Loop (HiL).

In an effort to reduce operational costs and increase efficiency, vehicle manufacturers are seeking alternatives to minimize the demand for prototype vehicles during the project development phase. The need for new validation methods that contribute to this has led to the use of computational models. Examples of these methods include the use of Software-in-the-Loop (SiL) with real-time simulations and automated testing routines.

This study aims to develop a methodology to demonstrate faults related to the diagnosis of the fuel pump electronic module (PEM) using a process with automated routines to allow some of the tests currently executed on prototype vehicles or HiL benches to be performed in a virtual environment (SiL).

INTRODUCTION

To stay competitive in the automotive industry, there is a growing need for increased efficiency and cost reduction. This is particularly important considering new regulations such as Euro 7, which demand additional development efforts and viable financial solutions. The collaboration between technology companies and the automotive industry is playing a crucial role in pushing the boundaries of the traditional market. [1].

The use of computational tools is essential for developments with shorter duration and lower costs, among these tests in virtual environments is the SiL (Software in the Loop), which involves simulating and validating the programming code to detect faults and improve code quality quickly and cost-effectively [2]. Typically, SiL testing and validation are performed in the early stages of the software development process.

This paper aims to demonstrate that fuel pump electronic module diagnosis, currently conducted on prototype vehicles or HiL benches can be performed and automated in the simulation environment (SiL), reducing validation time and costs and enhance the robustness of the OBD validation process.

Many companies in the automotive industry use existing modeling tools for product development, but in some cases, they create specific tools. An example of this is the Model.CONNECT and AVL Cruise, which were developed by AVL [3]. Ford, on the other hand, developed VMAPS (Vehicle Model Architecture Powertrain Simulation), implemented in Matlab/Simulink®, to simulate the vehicle's operation and its control systems using models. Another widely used software in the industry to automate testing routines is ECU-Test, developed by TraceTronic GmBH® [4]

To develop the process of validating faults monitored by the OBD system in a virtual environment (SiL), the following computer applications will be used: VMAPS, Matlab R2020b, ATI VISION, and ECU-Test.

As seen in Figure 1 below, in the SiL environment, the user only needs a computer with the applications mentioned at above paragraph to run the tests, while in HiL, the engine control module (PCM - Powertrain Control Module) and a HiL bench are required (investment around USD250,000.00).

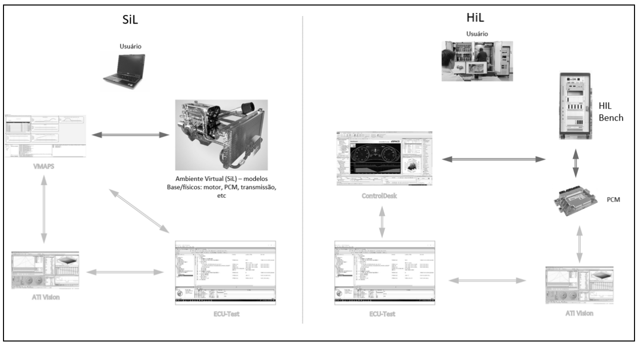

Figure 1. SIL x HIL Comparison

The scope of this work is to simulate the Diagnostic Trouble Codes (DTCs) related to the Pump Electronic Module (PEM) using VMAPS to simulate failures by modifying signals in the model associated with this component.

The Pump Electronic Module (PEM) has a driver which varies the voltage to the fuel pump through pulse width modulation. The voltage that Fuel Pump (FP) Driver Module delivers to the fuel pump is based upon a low frequency PWM input signal from the PCM. The FP module scales this input and provides an output duty cycle to the fuel pump. The associated PCM output is called Fuel Pump Command (FPC). The Fuel Pump Driver Module also maintains a diagnostic communication with the PCM through a low frequency, pulse width modulated (PWM) input signal called Fuel Pump Monitor (FPM).

Figure 2 shows the electric schematics among PCM, Fuel Pump Module (PEM) and Fuel Pump. FPC represents PCM output which controls fuel delivery to PEM while FPM represents PCM input which provides error states detected by PEM.

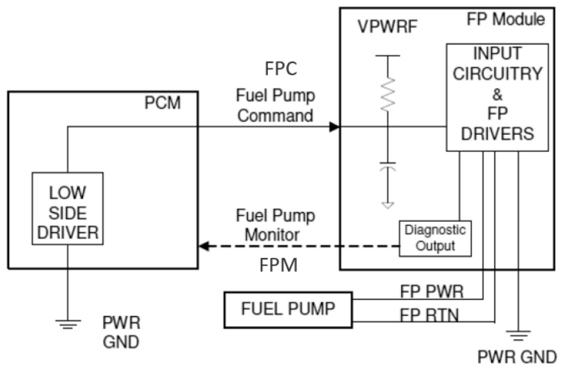

Figure 2. FPC and FPM Schematics

OBD DEMONSTRATION PROCESS

There are numerous OBD regulations implemented globally, each tailored to specific regions such as Brazil, Europe, and Japan. Notably, the state of California in the United States enforces one of the most stringent regulations worldwide through the California Air Resource Board (CARB) [5]. This regulation has been used as a key reference for the purposes of this paper. The OBD regulations contains detailed instructions about which engine components are required to be monitored and the types of faults that must be covered. This includes electrical faults and out of range monitors, as well as rationality checks for some subsystems.

The OBD regulation mandates that when a failure is identified during the initial driving cycle, a Diagnostic Trouble Code (DTC) is stored in a "pending" status. If the failure is detected again during a subsequent driving cycle, the DTC is then stored in a "confirmed" status. A drive cycle refers to a specific sequence of driving conditions that a vehicle must go through to complete a full diagnostic cycle. It typically includes a combination of various driving modes such as idling, acceleration, deceleration, and steady-state driving. When an emissions-related DTC is confirmed, it is also necessary that the “malfunction indicator lamp” (MIL) is lightened. After three completed drive cycles without the failure detected, DTC is healed. The MIL will be turned off at the beginning of the 4th drive cycle.

Each DTC must be demonstrated through a procedure that verifies for the correct fault detection, pending and confirmed status storage and healing as showed in Figure 3.

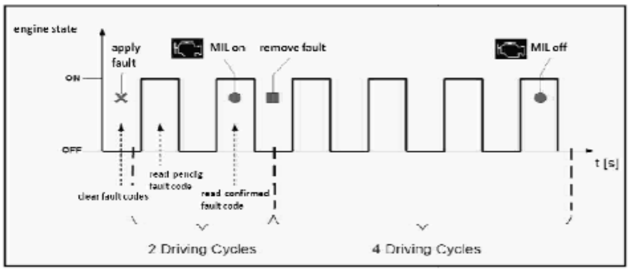

Figure 3. How a given diagnostic must behave in a vehicle/ MIL cycle

PUMP ELECTRONIC MODULE DIAGNOSTICS

There are different DTCs identified by PCM associated to PEM diagnosis. Some of them are caused by electric circuit faults and others related to PEM internal malfunctions. The pump module runs internal diagnostics and outputs the condition codes to the PCM via the FPM line as shown in Figure 2. Specific duty cycles, that are defined by the PEM supplier, report different fault modes which set distinct DTCs as described in the Table 1.

Fault Mode	Condition(s)	DTC
Normal Function	No faults detected	No DTC
Input Error from PCM	Invalid Duty Cycle Invalid Frequency	P025B
Internal PEM Error	Supplier specified	P064A
Output Error (pump driver)	Output Open (detected when PEM is commanded on)	P0627

Table 1. PEM Fault Modes and DTCs

A more detailed description of the DTCs related to the PEM are described below:

1) P025B - Fuel Pump Module "A" Control Circuit Range/Performance. It is detected when PEM receives an invalid input in the FPC channel. It will report back an invalid control signal to the PCM through FPM.

2) P0627 - Fuel Pump "A" Control Circuit Open. It is detected when PEM detects an error at the FP output (FP PWR and/or FP RTN). It will report that there is an Output Error in the fuel pump control circuit to the PCM through FPM.

3) P064A - Fuel Pump Control Module "A". It is detected when PEM recognizes an internal malfunction, it will report an Internal PEM error to the PCM through FPM.

4) P025A – Fuel Pump Module "A" Control Circuit Open. The fuel pump driver circuit will set a failure flag in the PCM strategy when a fuel pump control (FPC) circuit malfunction exists (open, short to ground or short to power).

5) U0109 – Lost Communication with Fuel Pump Control Module "A". When the PCM is no longer receiving edge transitions on the FPM signal (refer to Figure 4), meaning a duty cycle of 0% or 100%, a loss of communication failure is detected by the PCM. This DTC can also be set for intermittent loss of communication and can be caused by an open FPM circuit.

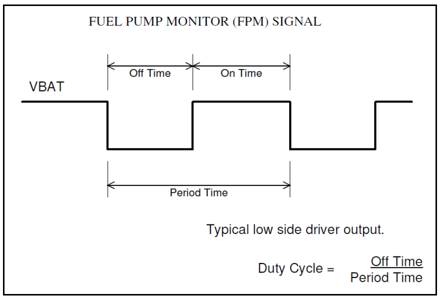

Figure 4. FPM Signal

EXPERIMENTAL SETUP AND MODELLING

Circuit faults can be simulated in VMAPS virtual environmental by modifying the signals in the model that correlates with physical pins at PCM hardware. The failures were applied through a test factor written in a .xml file. An application VMAPS package has a file called StrategyBusPCMFailure.xml that describes all the input and output signals mapped to the PCM pins, as well as the types of failure that each signal can detect.

In Figure 5 we can see the test factor written to insert an open circuit fault in the FPC circuit. The failure is inserted at Ms_FuelPump_Ctrl_mix (model correlation for FPC output pin) at the beginning of the test (startTime = 0) and applied for 100s (totalDurationSec = 100). This time is sufficient for PCM software to detect a fault in FPC circuit and set P025A DTC.

P025A_0A_FPC_Open.xml

```xml
<configuration name="SILSim">
  <platform>SIL</platform>
  <SILPlatform>
    <processes>
      <process name="PCM">
        <electrialFailureConfigs failureConfigFilename="..\..\Controllers\Modules\PCM\IO\StrategyBusPCMFailure.xml">
          <electrialFailureConfig startTime="0" updateRate=".1">
            <failureSets>
              <failureSet>
                <failures>
                  <failure signalName="Ms_FuelPumpCtrl_mix" failureType="Open" totalDurationSec="100"/>
                </failures>
              </failureSet>
             </failureSets>
          </electrialFailureConfig>
        </electrialFailureConfigs>
      </process>
    </processes>
  </SILPlatform>
</configuration>
```

Figure 5. Test factor to simulate FPC open circuit fault

Figure 6 describes how the test factor was written to insert an intermittent open circuit fault in the FPM circuit. Failure is inserted at Ms_FuelPumpMontr1_LowPres_mix (model correlation for FPM input pin). The "Open" failure is applied for 2s for a given period (totalDurationSec = 2) in a way such the failure is inserted for 2s, and then removed for 2s repeatedly throughout the simulation. This failure method makes PCM software to set U0109 DTC.

U0109_1F_FPM_Intermittent.xml

```xml
<configuration name="SILSim">
  <platform>SIL</platform>
  <SILPlatform>
    <processes>
      <process name="PCM">
        <electrialFailureConfigs failureConfigFilename="..\..\Controllers\Modules\PCM\IO\StrategyBusPCMFailure.xml">
          <electrialFailureConfig startTime="0" updateRate=".1">
            <failureSets>
              <failureSet>
                <failures>
                  <failure signalName="Ms_FuelPumpMontr1_LowPres_mix" failureType="Open" totalDurationSec="2"/>
                </failures>
              </failureSet>
             </failureSets>
          </electrialFailureConfig>
          <electrialFailureConfig startTime="4" updateRate=".1">
          <failureSets>
            <failureSet>
              <failures>
                <failure signalName="Ms_FuelPumpMontr1_LowPres_mix" failureType="Open" totalDurationSec="2"/>
              </failures>
            </failureSet>
          </failureSets>
          </electrialFailureConfig>
          <electrialFailureConfig startTime="8" updateRate=".1">
            <failureSets>
              <failureSet>
                <failures>
                  <failure signalName="Ms_FuelPumpMontr1_LowPres_mix" failureType="Open" totalDurationSec="2"/>
                </failures>
              </failureSet>
            </failureSets>
          </electrialFailureConfig>
          <electrialFailureConfig startTime="12" updateRate=".1">
            <failureSets>
              <failureSet>
                <failures>
                  <failure signalName="Ms_FuelPumpMontr1_LowPres_mix" failureType="Open" totalDurationSec="2"/>
                </failures>
```

Figure 6. Test factor to simulate FPM open circuit intermittent fault

To simulate PEM internal diagnosis, it was necessary to create a block in the model to override modelled FPM signal and simulate the specified duty cycle for each of DTC. Three model variables were created:

- MIP_Dc_FuelPumpMontr1DtyCyc_LowPres_norm.B_Ovrrd_bit: switch to override FPM signal. It is used to remove the FPM control from the model. When set to true (value 1), FPM can be overridden.

- MIP_Dc_FuelPumpMontr1DtyCyc_LowPres_norm.Z_Gain_uu: gain applied to FPM signal. For this study the value was set to 0.

- MIP_Dc_FuelPumpMontr1DtyCyc_LowPres_norm.Z_Offst_uu: offset applied to FPM signal. It is used to manually set the emulated FPM PWM signal from PEM to PCM.

The resulting model equation is below, and an example of a block diagram that represents this equation can be observed on Figure 7.

Overridden Signal = Gain * ModelGeneratedSignal + Offset
Equation 1. Override model equation

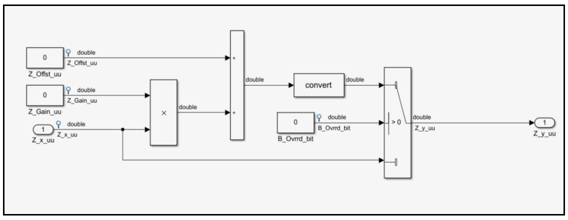

Figure 7. Gain Offset Block Diagram example

As example, to validate a given DTC it is necessary to set FPM PWM to 20%, so the parameters shall be set as:

- MIP_Dc_FuelPumpMontr1DtyCyc_LowPres_norm.B_Ovrrd_bit = 1

- MIP_Dc_FuelPumpMontr1DtyCyc_LowPres_norm.Z_Gain_uu = 0

- MIP_Dc_FuelPumpMontr1DtyCyc_LowPres_norm.Z_Offst_uu = 0.2

Figure 8 shows a test case created in VMAPS GUI (Graphical User Interface) to override the PWM signal sent through FPM to the PCM. The failure is inserted using the MIP_Dc_FuelPumpMontr1DtyCyc_LowPres variables as described before. The test case has a Timed Stimuli block where the switch to override FPM signal is set to 1 at the beginning of the test and it is applied for 100s (Segment Duration = 100). This time is sufficient for PCM software to detect the fault and set a given DTC.

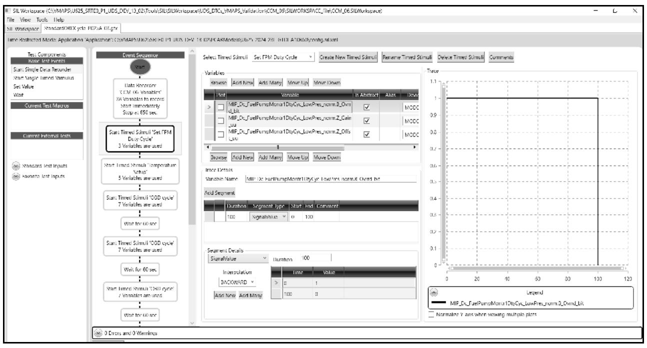
Figure 8. Test case to simulate PEM fault

RESULTS AND DISCUSSION

To execute the DTC Demonstration procedure in VMAPS virtual environment, test cases were created to replicate the complete MIL cycle for each DTC related to Fuel Pump Diagnostics. For these DTCs, a drive cycle that consists in key on, engine crank followed by 30s in idle, and subsequent key off for 20s is sufficient for PCM to run the monitor and store the DTC status.

Running a test case built with drive cycles mentioned before associated with the test factor from Figure 5, the achieved result can be observed in Figure 9. This test factor applied an open circuit fault in the FPC circuit for 100s, which prevents the engine from cranking at the initial 2 drive cycles as can be noted in the upper graph which contains the ENGINE_SPEED trace. This happens because the PCM doesn't receive the fuel pump command and therefore is not able to inject fuel to raise the engine speed above the starter motor rotating speed.

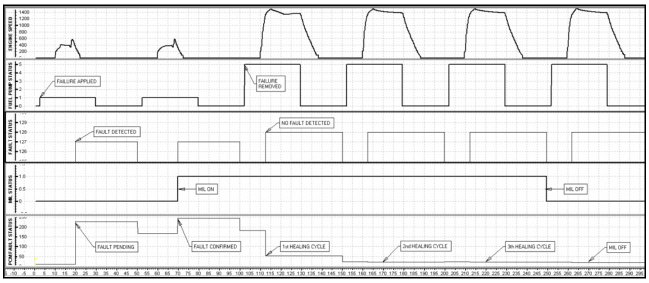

Figure 9. P025A DTC Demonstration Cycle

Similarly, when the test factor from Figure 6 is linked to the test case to reproduce the DTC Demonstration procedure the outcome is the data found on Figure 10. In this case the engine cranks and engine speed increases even with the failure applied. And since the failure is inserted in an intermittent manner, the variable that shows the fuel pump failure status (FUEL_PUMP_STATUS) in the second graph keeps changing its value during the 100s while the failure is inserted.

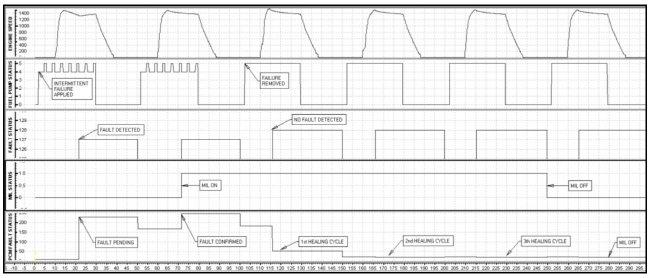

Figure 10. U0109 DTC Demonstration Cycle

For the PEM internal diagnostic DTCs the failure is simulated directly in the test case, without the need of a test factor, through a Timed Stimuli block as highlighted in Figure 8. The results of the test case with PWM duty cycle signal related to the P0627 DTC is displayed in Figure 11. P025B and P064A follow same methodology as explained above and can be simulated by changing the offset value mentioned in Equation 1. For this paper, only P0627 is being demonstrated.

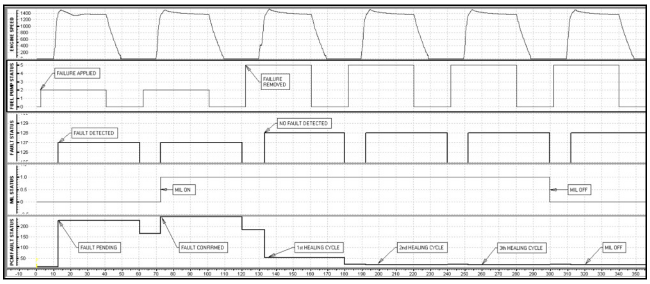

Figure 11: P0627 DTC Demonstration Cycle

Summarizing, the following results can be drawn from the tests carried out in VMAPS Workspace and exhibited on Figures 9, 10 and 11:

- All failures applied were detected by the model. The second graph shows that the variable related to the fuel pump diagnostics strategy (FUEL_PUMP_STATUS) detects the simulated failure and communicates the fault present status to the diagnostic executive feature after a calibratable amount of time when the FAULT_STATUS changes in the third graph.

- The fourth graph is related to the MIL_STATUS and it can be observed that it turns on at the second drive cycle with the failure applied and detected, and then at the 4th drive cycle after the failure is removed the MIL is off.

- The fifth graph displays the DTC KAM record for PCM_FAULT_STATUS, which indicates if the DTC is pending, confirmed, or healing.

SUMMARY / CONCLUSIONS

The SiL testing strategy proposed in this paper was proven to be robust and efficient. Transfer the validation to a simulation environment reduces the number of physical tests (requiring a PCM Hardware or/and Prototype Vehicle) and contributes to zero prototype mission, a goal for all vehicle manufactures.

All alternative solutions, comparable to the one this paper proposes, are very helpful to increase competitiveness during product development. Since Hardware is not required, SiL Tests can anticipate issues that would only be discovered late in development when physical parts are available. As defined by James Folkestad and Russell Johnson, cost of design changes significantly increases at late development phases [6].

The validation of diagnostic functions plays a fundamental role in software development, requiring the simulation of failures. Depending on the system in use, different methods of failure simulation and tool chains can be employed. However, the approach presented in this study offers the advantage of reusing failure simulation methods across various applications, eliminating the need for system dependency [4]. With the implementation of this system, a comprehensive validation of all diagnostic trouble codes (DTCs) associated with fuel pump diagnostics has been successfully executed. The notable improvement in testing speed has enabled more frequent and earlier validation of these monitoring functions throughout the development process.

Moreover, it is possible to accomplish an initial version of the mandatory validation for On-Board Diagnostics functions in the early stages of development by using a system that closely mirrors the final production vehicle. Many components of the method described in this study are already widely employed in diverse software testing fields. However, the integration of these elements into a fully automated tool represents a notable advance in the testing of automotive monitoring functions.

The method presented at this study can be replicated for other OBD monitors with adjustments to the model, test factors and test cases to simulate failures in other sensors and actuators connected to Powertrain Control Module. Replication for other monitors will be subject of a future study at the company.

REFERENCES

[1] KENDAL, I. R.; JONES, R. P. An Investigation into the Use of Hardware-in-the-Loop Simulation Testing for Automotive Electronic Control Systems. Control Engineering Practice 7, 1343-1356, 1999.

[2] CAMERETTI, M C; LANDOLFI, E; TESONE, T; CARACENI, A. Virtual Calibration Method for Diesel Engine by Software in The Loop Techniques. International Journal of Automotive and Mechanical Engineering; Kuantan Vol. 16, 2019.

[3] BALIC, J. "Plataforma para desenvolvimento virtual e validação de aplicações em Direção Autônoma e RDE", p. 162-177 . In: XXVII Simpósio Internacional de Engenharia Automotiva. São Paulo: Blucher, 2019.

[4] RICHERT, F., BRÜCKNER, D., "Automated Validation Process for On-Board-Diagnosis Functions", IFAC Proceedings Volumes, Volume 43, Issue 7, p. 727-732, 2010.

[5] CARB: California Air Resource Board (2007). *Final regulation Order for OBD II and Emission Warranty Regulations, Section 1968.2. Approved on November 9, 2007.* OBD II regulations and Rulemaking: https://ww2.arb.ca.gov/resources/documents/obd-ii-regulations-and-rulemaking

[6] JOHNSON, R; FOLKESTAD, J. Resolving The Conflict Between Design And Manufacturing: Integrated Rapid Prototyping And Rapid Tooling (IRPRT), 2001

DEFINITIONS / ABBREVIATIONS

DTC Diagnostic Trouble Code

FP	Fuel Pump
FP PWR	Fuel Pump Power
FP RTN	Fuel Pump Return
FPC	Fuel Pump Command
FPM	Fuel Pump Monitor
GUI	Graphical User Interface
HiL	Hardware in the Loop
KAM	Keep Alive Memory
MIL	Malfunction Indicator Lamp
OBD	On Board Diagnosis
PCM	Powertrain Control Module
PEM	Pump Electronic Module
PWM	Pulse Width Modulation
PWR GND	Power Ground
SiL	Software in the Loop
VMAPS	Vehicle Model Architecture Powertrain Simulation